

AD-A042 897

ILLINOIS UNIV AT URBANA-CHAMPAIGN CENTER FOR ADVANCED--ETC F/G 5/2  
RESEARCH IN NETWORK DATA MANAGEMENT AND RESOURCE SHARING. THE E--ETC(U)  
FEB 76 G G BELFORD, P M SCHWARTZ, S SLUIZER DCA100-75-C-0021

UNCLASSIFIED

UIUC-CAC-DN-76-181

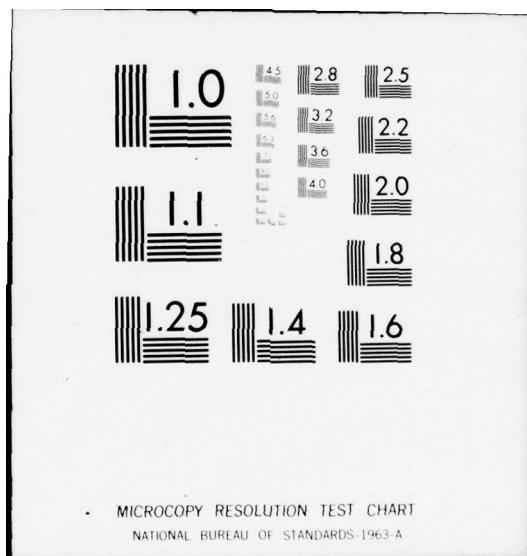
CCTC-WAD-6501

NL

1 of 1  
ADA042897



END  
DATE  
FILED  
9-77  
DDC



AD A 042897

# Center for Advanced Computation

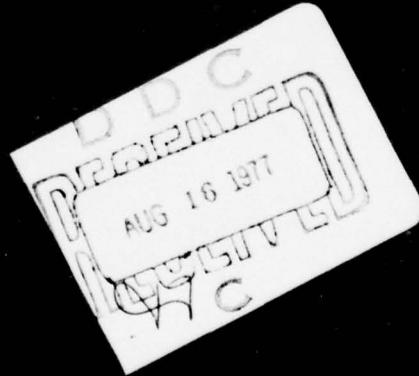
3  
B.S.

**CAC Document Number 181  
CCTC-WAD Document Number 6501**

*Research in  
Network Data Management and  
Resource Sharing*

**The Effect of Backup Strategy on  
Data Base Availability**

February 1, 1976



**DISTRIBUTION STATEMENT A**  
Approved for public release;  
Distribution Unlimited

CAC Document Number 181  
CCTC-WAD Document Number 6501

Research in  
Network Data Management and  
Resource Sharing

The Effect of Backup Strategy on  
Data Base Availability

by

Geneva G. Belford  
Paul M. Schwartz  
Suzanne Sluizer

Prepared for the  
Command and Control Technical Center  
WWMCCS ADP Directorate  
of the  
Defense Communication Agency  
Washington, D.C.

under contract  
DCA100-75-C-0021

Center for Advanced Computation  
University of Illinois at Urbana-Champaign  
Urbana, Illinois 61801

ACCESSION for		
THIS	White Section <input checked="" type="checkbox"/>	
DOC	Buff Section <input type="checkbox"/>	
ANNOUNCED <input type="checkbox"/>		
J. STIFICATION _____		
BY _____		
DISTRIBUTION/AVAILABILITY CODES ... and/or SPECIAL		
A		

February 1, 1976

Approved for release:

*Peter A. Alsberg*  
Peter A. Alsberg, Principal Investigator

(See 1473)

## TABLE OF CONTENTS

	Page
Executive Summary . . . . .	1
The problem . . . . .	1
The model . . . . .	2
Conclusions . . . . .	2
Introduction . . . . .	4
File allocation . . . . .	4
Network reliability modeling . . . . .	5
Modeling computer system reliability . . . . .	6
Modeling backup and recovery strategies . . . . .	6
The present work . . . . .	7
The Model . . . . .	9
Overview . . . . .	9
Parameters . . . . .	11
Single-copy availability . . . . .	11
Discussion of the parameter k . . . . .	12
Availabilities for two backup strategies . . . . .	13
Experiments and Discussion . . . . .	17
Remote journaling . . . . .	17
Frequently updated remote journal . . . . .	20
Running spares . . . . .	21
Effect of varying Y . . . . .	22
Conclusions . . . . .	23
References . . . . .	24
Appendix 1: Extension of Chu's Formula . . . . .	25

TABLE OF CONTENTS (continued)

	Page
Appendix 2: Stochastic Considerations . . . . .	27
Basic assumptions . . . . .	27
Probability that backup fails before master is ready	
for use . . . . .	28
Probability that backup site fails before its copy	
is ready . . . . .	32
Appendix 3: Time to Process the Audit Trail . . . . .	34
Elementary analysis . . . . .	34
Analysis using queueing theory . . . . .	34
Appendix 4: Sensitivity to Parameter Values . . . . .	39

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED <input type="checkbox"/>	
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	SPECIAL
P	

### Executive Summary

The problem. The availability of a data base may be simply defined as the fraction of time that the data is available to users. Many things can cause a data base to become unavailable in a network setting. If the data base is stored at the same location as the user, the system through which the data must be accessed may fail, or the device on which the data base is resident may crash. If the data base is located at a remote site on the network, the remote site or system may fail, the network may partition so that the remote site cannot be reached, or some local failure may make the network inaccessible to the user.

In most of these cases, availability can be considerably improved if a backup copy of the data base exists. If copies of the data base exist at two sites in the network, the danger of losing access because of network partitioning or site failure is reduced. Furthermore, if a local device holding all or part of the data base crashes, data may be destroyed. It is likely to be much faster (as well as more reliable) to ready a locally archived backup copy of the data for usage than to try to recover the lost or degraded data from audit trails, etc.

How much the existence of a backup copy improves availability depends on a number of factors. For example:

- 1) How available is the backup copy? (Is it stored on disk for immediate access? If it is stored on tapes, a sizeable delay may be incurred while the tapes are located, mounted, and loaded onto a rapid-access device.)
- 2) How up-to-date is the backup copy? (Are all updates applied to the backup copy as rapidly as possible? Is there a long backlog of updates that must be processed before the data base is really ready for use?)

3) How often is the site (or device) holding the data base likely to fail? (If failures are infrequent, the backup copy may provide little improvement in availability.)

Even small improvements in availability can, of course, be important. Availability can be over 0.99 and still be disastrously low if, say, the data is unavailable for one 24-hour period during a year and that period happens to be during a crisis. It is important, therefore, to understand thoroughly how availability is affected by the factors discussed in the preceding paragraph, and hence by the strategy used for backing up a data base.

The model. We have developed simple algebraic formulas for availability as a function of the factors listed above. Additional parameters are incorporated to model the delay incurred in initiating the process of readying the backup copy, the rate at which updates are generated, and the rate at which updates are processed. We have assumed the existence of a single backup copy, and have studied the improvement in availability that the existence of a backup provides over single-copy availability. The formulas are kept simple by using average values for parameters that are actually random variables. For example, we use the "mean time between failures" in the availability formula, while system failure is actually a random process. In appendix 2, we look into the validity of this simplification and conclude that its affect on computed availabilities is, in most realistic situations, to make them appear only slightly larger than they actually would be.

Conclusions. One main conclusion from studying the model is that a backup copy can improve the availability of a data base by as much as 5 to 10 per cent. To put this result into more concrete terms,

suppose that a single copy is likely to be down for two hours per day (availability = .917). A 5 percent improvement would produce an availability of .963, or a reduction of probable down time to about 54 minutes.

A second important conclusion is that if the backup copy is readily accessible and kept reasonably up to date, the availability is very close to 1. On the other hand, if the backup copy is stored on tape, so that it is relatively out of date and locating it is a time-consuming process, availability may be little better than was provided by a single copy. (This is because one can probably repair the original system about as rapidly as one can ready the backup.) Indeed, a backup of this sort tends to be mainly useful for recovery from some accident which destroys data in the original data base.

In this study, we have necessarily restricted ourselves to trying to answer a few specific questions and to computing availabilities for only a limited number, or range, of parameter values. However, the formulas we have developed - and, even more, the simple, straightforward approach which yielded those formulas - have applicability in a wide variety of settings. The most important aspect of this work is not the particular numbers or formulas obtained but the tools developed for studying availability in general. With little additional effort, these tools can be used to provide answers to other questions regarding the effect of backup strategy on availability.

## Introduction

We here use the term availability to mean the fraction of time that a data base is available to respond to user requests or queries. In any setting, and particularly in a network, availability is a function of the reliability (or availability) of many components - host computers, network communications lines, etc. - as well as of strategies for backup and recovery. In this section we first discuss some of the past modeling research that has yielded results relevant to database availability, and then introduce the line of work which we have pursued.

File allocation. One of the factors to be taken into account in distributing copies of a file to various network sites is the number of copies needed for an acceptable degree of availability. Chu [1973] takes account of this factor in the following way. First, he defines the availability of a piece of equipment (e.g., communication line or computer) as

$$\text{Availability} = \frac{F}{F + X},$$

where  $F$  is the mean time between failures and  $X$  is the mean time to repair. Then, assuming

- 1) all computers in the network have identical availability  $A$ ,
- 2) all communication channels have identical availability  $c$ , and
- 3) the network is completely connected;

Chu obtains the following formula for the availability of the  $j$ th file:

$$A(1 - (1 - Ac)^{r_j}),$$

where  $r_j$  is the number of copies of the  $j$ th file in the network. Once  $A$  and  $c$  are known, it is a simple matter to choose  $r_j$  so as to bring the availability of a remote copy up to a satisfactory level. Overall availability, however, is bounded by the factor  $A$ , the availability of the

requesting computer, which is apparently assumed not to possess a copy of the file.

Although Chu's model, with its assumption of complete homogeneity of network components, may seem oversimplified, an analogous analysis can be readily carried out in the heterogeneous case to yield only slightly more complex expressions. (See appendix 1.) Notice, however, that this model presents another problem. It implicitly assumes that the files are static, or are simultaneously kept up to date by some trouble-free process. In fact, the development of algorithms to keep segments of a data base identical (or nearly so) is a topic of current research. (See the chapter on Automated Backup in CAC Doc. No. 162, JTSA Doc. No. 5509.)

Network reliability modeling. Another simplification in Chu's model is the assumption that a direct communication line connects every pair of sites. This assumption allows Chu to use a single parameter to describe availability of a link from one site to another. In a general network, this availability will depend in a complex way upon network topology. Several alternate paths may exist between two given sites. Each of these paths may involve more than one "hop" and so more than one piece of subnet hardware. Indeed, in the ARPA network it has been found that the failure rate for IMP's is about the same as that for communication channels, and that IMP failures therefore have the more drastic effect on communications reliability [Frank, Kahn, and Kleinrock, 1972]. Graph theoretical techniques for computing availability from component reliabilities are, however, well known. The paper by Frank et al. contains a brief review of these techniques. No great difficulty is envisioned in applying them to any given network (such as the WIN) to obtain

host availabilities. These may then be used in the formula given in appendix 1 to obtain rough estimates of file (or data base) availability.

Modeling computer system reliability. Another parameter in Chu's model that requires more detailed analysis for complete understanding is computer availability. One source of information on computer availability is direct system measurement. On a lower level, however, failures can be modeled to yield, in addition to overall figures on expected system reliability, useful insights into repair and backup strategies.

Borgerson and Freitas [1975] recently published a fairly detailed stochastic model for computer system failure. Their model is based on four distinct causes of crashes and their interrelationships. Their ultimate result is a formula giving the probability density for the event that the system crashes due to a failure. For our availability analysis, however, there seems to be little need to include this level of detail; we are simply concerned with failure rate - a measurable quantity.

Modeling backup and recovery strategies. The discussion above has been limited to availability questions involving network and site reliabilities. On a lower level, the data base itself may "crash" or may acquire errors. It is important that strategies for returning a data base to its correct state be devised and studied.

A recent paper [Chandy et al., 1975] provides models for rollback and recovery strategies. These strategies run as follows. At certain points in time (checkpoints), a copy of the data is made and stored. A listing of subsequent data updates (i.e., an audit trail) is then kept. When the master data base fails, it may then be recovered by beginning with the old copy from the checkpoint and using the audit trail to bring it up to date. Chandy et al. use queueing theory to

model the processing of the audit trail. From the expected time to complete this process, they can compute the total recovery time. The length of the audit trail, and hence the time to recover, is a function of the time interval between checkpoints. Optimization of availability with respect to intercheckpoint time can then be carried out. Models of some complexity are developed which take into consideration the possibility of errors during recovery and the possibility of a transaction arrival rate which varies in a cyclic manner (as opposed to being constant). The results appear to be very useful for developing insights into recovery strategies, particularly for single-site systems. In a network environment, however, it may be reasonable to assume that the backup copy is stored remotely. In this case it does not make sense to assume that the data is always restored from the backup, because of the long time required to transfer a data base through the network. The strategy then is to transfer the queries to the available copy.

The present work. In this note we attempt to quantify the improvement in data base availability which can be achieved by storing a backup copy at one (or more) remote sites in a network and transferring usage to the backup when the master fails. We also discuss the practicality of certain alternative management strategies.

To simplify the analysis, we will not consider various possible causes of data base failure, but will assume that the data is available when the host computer is running and is available (if remote) by way of the network. We will therefore not be considering a detailed analysis of the type of Borgerson and Freitas, nor will we be concerned with network reliability modeling. Host failures are so much more common than communications link failures that the latter can be neglected in our simple model.

Furthermore, we will not take into account scheduled down time of the host computer, on the assumption that if down time is scheduled, transfer to a backup copy is automatic and immediate, and leads to no loss in availability. The very existence of a backup copy at an alternate network site will of course improve availability considerably over the case where only one site has a copy. Indeed, Chu's model (or a simple modification of it) can be used to determine the improvement in availability due to multiple copies when all copies are equally usable. Since some readers may find this question of interest, we have included a discussion of it in appendix 1.

### The Model

Overview. The process we are modeling may be described as follows. Several sites in a network possess copies of a data base. One of these copies is designated as the master copy. The others are referred to as spares or backups. All queries for the data base are sent to the master site (i.e., the site holding the master copy). Updates are applied to the master copy as soon as possible after they are generated, so that the master copy is kept up to date. Two basic strategies for updating the spares are encompassed by our model:

- 1) Running spares. Spares are updated almost as rapidly as the master.
- 2) Remote journaling. Up-to-date copies of the data base are periodically sent to the backup sites for storage. In between this periodic journaling, updates are logged in an audit trail for application to a spare if and when one is needed.

Occasionally the master copy becomes unavailable. We assume this is caused by a failure of the host possessing the master copy and not by, say, communication line failure. When the master site fails, some sort of communication among sites takes place to determine which of the spares should take over the responsibility of being the new master. The length of the time interval from when the old master fails to when the new master is decided upon is assumed to be a fixed constant.

Once a new master site has been selected, the spare copy at that site must be readied to receive queries. This process of getting the backup ready may involve time-consuming operations such as loading the data from tape and processing the audit trail of updates which have not yet

been applied to the backup copy. How close to "ready" the backups should be kept is another strategy question which may be studied by our model.

As soon as the old master fails, the process of repairing it begins. After the host has been repaired, the data base itself must be readied. A backlog of updates has been accumulating while the master was being repaired, and these updates must be applied to the data.

Thus, after a certain time lapse, the old master (i.e., the primary master) is again ready to receive queries. The question is, should we immediately reinstate the old master, or should we continue to send queries to the new master until it fails? With our model we can study the impact on availability of how we answer this question. There may, however, be other issues involved. For example, most of the queries to the data base may originate at the primary master site. In this case there are cost and/or response advantages to be gained by transferring usage back to the primary site as soon as possible.

For simplicity, we have described the process we are modeling in fairly specific terms. It should be noted, however, that little change is needed to model other, similar processes. For example, the backup and master copies may be located at the same site. And the failures we are concerned with may be the crashing (with accompanying data destruction) of the device holding the master copy. In this case there are no network messages to transfer usage to a remote site, nor need we worry about repairing the host. But the need to get the backup ready by loading the copy and then bringing it up to date remains the same. Only trivial changes in the availability formulas we have derived will allow us to study this sort of closely related process.

Parameters. The parameters in our model are as follows:

F = mean time between computer failures, assumed to be the same for all host computers.

X = expected time to repair computer.

L = expected time to load the data base copy at the remote site.

Y = time that the audit trail of updates has been growing (i.e., time since the copy was correct).

k = the ratio of update arrival rate to update processing rate.\*

D = time delay between when the master fails and when the remote site determines this fact and starts to get its copy ready for use.

Single-copy availability. First, consider the case where there is a single copy of the data base. The availability of this copy is then

$$A_o = \frac{F}{F + X + kX}.$$

This is the usual formula for availability (mean time between failures divided by mean time between failures plus mean time to recover). The mean time to recover includes repair time X plus the time kX to process the updates accumulated while repairs were made. (This formula for recovery time is that used by Chandy et al. [1975].) There is a question as to whether the term kX should be included here, since the site is technically "up" after time X. But in a network setting, it does seem appropriate to assume that updates initiated at remote sites

---

\*The parameter k is referred to in the literature as a "compression" factor [Chandy et al., 1975]. This is not to be confused with the usual data compression factor which indicates by how much data is compressed for storage or transfer.

are being logged somewhere, so that there does exist an update list to be processed. In addition, we are interested primarily in comparing  $A_0$  with availabilities computed for multi-copy strategies, where the copies are assumed to be up to date.

Discussion of the parameter k. The rationale for using the formula  $kY$  for the time to process an audit trail that has been accumulating for a time period of length  $Y$  is as follows. Suppose  $u$  is the rate of arrival of updates, and  $b$  is the rate of processing them. Then during time  $Y$  a total of  $uY$  updates have accumulated and it takes time  $uY/b = kY$  to process these. (We have defined  $k = u/b$ .) However, the system can not really be said to be caught up after this much time, since more updates were accumulating while the backlog was being processed. Let us define  $T_c$  as the catch-up time, or time for the system to catch up after a backlog of updates has accumulated. The determination of an appropriate expression for  $T_c$  turns out to be a nontrivial problem. This problem is examined in detail in appendix 3. We find there that for a reasonable range of values of  $k$ ,  $2kY$  may be a more appropriate expression for  $T_c$  than is  $kY$ .

In the remainder of this note, however, we will consider  $k$  as an effective proportionality constant, defined by the assumption that  $kY$  is the time to catch up after updates have been accumulating for time  $Y$ . The reader should keep in mind that then  $k$  is not equal to  $u/b$  but is somewhat larger, perhaps by as much as a factor of 2 or more. It is, of course, possible for a site to obtain an effective  $k$  by measurement. A  $T_c$  can be measured as the length of time between the time when processing of the update backlog begins and when the update queue is first noted to be empty. An average over several observations of  $T_c/Y$  should yield an acceptable value for the effective  $k$ .

Availabilities for two backup strategies. We shall consider two strategies for transferring usage back and forth between master copy and backup copy. Strategy 1 runs as follows. After the master copy is determined to have failed, the remote copy is then brought up (after a time lapse of  $D + L + kY$ ) and usage is transferred to it. Meanwhile the old master is being repaired. Queries and updates are sent to the new master, however, until it fails, at which time the process repeats: another "new" master is identified and activated. (This may or may not be the "old" master.) Since the remote site may have been up for some time since its last failure, one might think that, after the new master site is identified, time until failure is only  $F/2$ . This is only true, however, if the time between failures is always precisely  $F$ . If, as we are assuming, failures form a Poisson process (i.e., occur randomly) it may be shown that the expected time until failure is not  $F/2$  but  $F$ . (See, for example, [Kleinrock, 1975, pp. 169-174].) This result, known in renewal theory as the "paradox of residual life", may be explained intuitively as occurring because the old master has a higher probability of failing during a relatively long inter-failure period at the new site.

Strategy 1 is diagrammed in figure 1. Looking at the diagram and ignoring the initial time period, one can see that the fraction of time some copy of the data base is available is

$$A_1 = (F - L - kY)/(F + D).$$

The quantity  $A_1$  is then the data base availability under strategy 1.

Notice also that an obvious built-in assumption can be read from the figure.

$$(1) D + L + kY < X + kX$$

If this inequality is not satisfied, it theoretically does not pay to

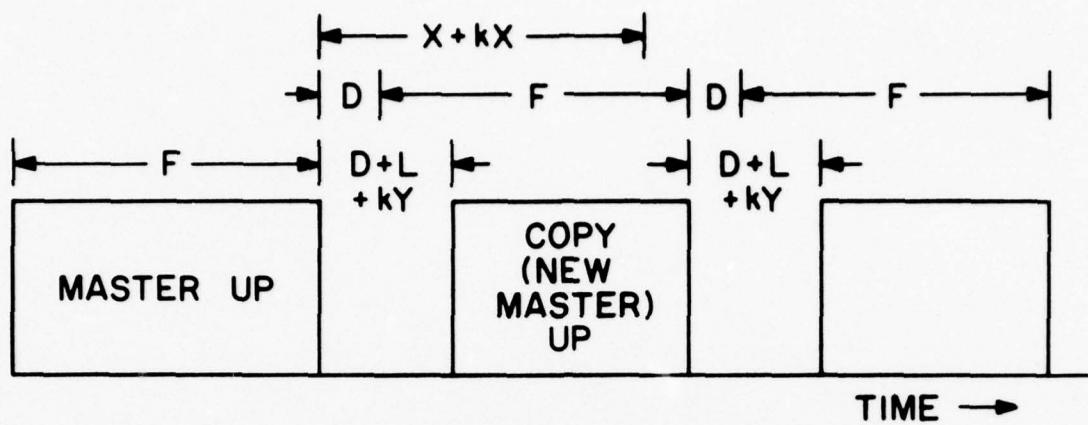


Figure 1  
Diagram of strategy 1.

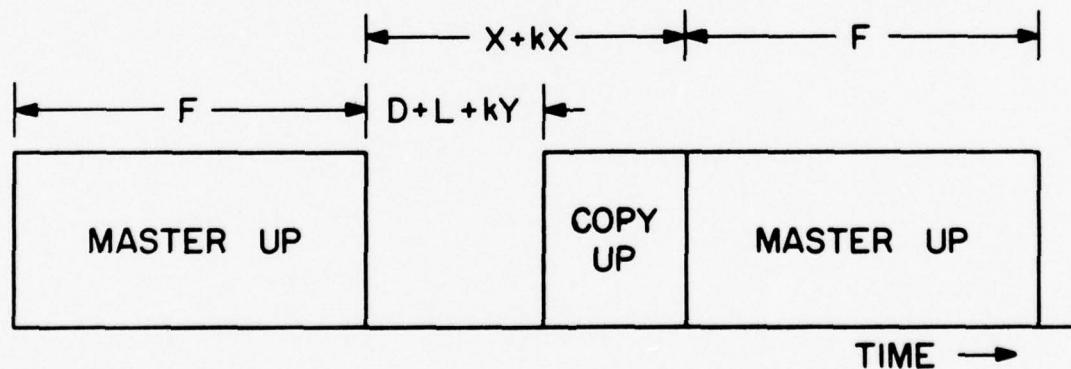


Figure 2  
Diagram of strategy 2.

store a remote copy, since the master is expected to be repaired and updated before the remote copy can be activated.

Strategy 2 is to immediately replace the copy by the old master as soon as the latter has been brought back up. This scheme is diagrammed in figure 2. Again, inequality (1) must hold in order for the diagram to be meaningful, and the availability formula can be read from the diagram:

$$A_2 = 1 - \frac{D + L + kY}{F + X + kX}.$$

By looking at the ratio  $A_2/A_1$ , one can easily show that as long as D is small compared to the other parameters (a realistic assumption)  $A_2$  is always greater than  $A_1$ . That is, strategy 2 is the better strategy, as one might intuitively infer from comparison of figures 1 and 2. In the following sections we will therefore restrict consideration to strategy 2.

There are two additional assumptions which must be made in order for our model of either strategy to be valid. One assumption is that  $D + L + kY$  is sufficiently small compared to F that there is little likelihood of a failure of the remote host during the recovery process. In addition, we assume that there is a negligible probability that the copy may fail before the master is again ready. If either of these assumptions is false, availability will generally be less than what we compute from our model. A probabilistic analysis of these assumptions is contained in appendix 2. Notice that strategy 2 is a two-copy strategy. Transfer of usage back and forth between the primary site and a single backup is specifically modeled. In strategy 1, however, after the backup fails, usage may be transferred to a third copy instead of to the copy at the primary site. Thus in this strategy, even if the new master

is likely to fail before the old one is again ready, the model is not invalidated as long as a second backup is available.

In the experiments to be discussed in the next section, we have ignored probabilistic considerations. The reader should simply keep in mind that availabilities are always slightly less than we compute there. The quantities that we investigate are:

- 1)  $A_2$ , the availability under strategy 2, and
- 2)  $I$ , the improvement in availability due to the existence of a backup copy.

That is,

$$I = \frac{A_2 - A_0}{A_0} = \frac{X - D - L + k(X - Y)}{F}.$$

### Experiments and Discussion

Remote journaling. In order to model a remote journaling process, we assume that the parameter  $Y$  is large; for simplicity we assume that it is equal to  $F$ . Thus we are essentially assuming that, whenever the master comes up after a failure, a copy of the up-to-date data base is shipped off to any remote site which contains a copy of the data base. (Or that the remote data base, having been used as a master copy while the master was down, already possesses an up-to-date copy at this time.)

It is interesting to note that journaling remotely by shipping the data base over the network is not feasible on a regular basis. For example, consider a data base of  $4 \times 10^7$  bytes (roughly FORSTAT size). At a network throughput of 15 kilobits per second (faster than normal for the ARPANET), it would take approximately 6 hours to ship a data base of this size. Daily backup by, say, sending tapes by courier would, however, be feasible in many situations.

The data copy at the remote site will be generally assumed to be on tape. The value  $L = 0.5$  hr. has been assumed in the computations since it is approximately the time to read two to three tapes. The parameter  $D$  is probably on the order of one or two seconds, but we have taken it to be .01 hr. as an absolute upper bound.  $X = 1$  hr. seems to be a reasonable value for repair time. With these parameters, we get the following formula for improvement  $I$  in availability as a function of  $F$  and  $k$ .

$$I = \frac{A_2 - A_0}{A_0} = \frac{0.49 + k(1 - F)}{F}$$

It is difficult to estimate what a reasonable value of  $k$  should be. In a similar analysis, Chandy et al. [1975] suggest that  $k$  should be 0.1 or

less. Clearly the value will depend on the usage pattern for the data base; we have already discussed how it may be measured for a real system. However, notice that, with  $k = 0.1$ , inequality (1) states that

$$.5I + 0.1F < 1.1.$$

Hence for this large a  $k$  the time to process the audit trail is so long that, without taking into account stochastic considerations, the master is able to get ready before the backup copy whenever  $F > 5.9$  hrs. This is an unreasonably low value. Furthermore, we show in appendix 2 that for these values of  $D$ ,  $L$  and  $k$ , and for all values of  $F$  (with  $Y=F$ ), there is a better than 10 percent chance that the backup site fails before its copy can be gotten ready. In short, we are unlikely to adopt a remote journaling strategy in these circumstances.

To get a feel for the value of remote journaling in a case when it may be practical, we therefore take  $k = .01$ ; i.e., we assume that there are few updates. Inequality (1) then restricts the model to  $F < 50$ . A graph of  $I$  vs.  $F$  in this case may be seen in figure 3. Values of  $A_o$  have also been plotted in the figure for reference. Notice that for reasonable values of  $F$  the improvement in availability is less than 5 percent. If  $A_o$  is low, this may not be enough to make remote journaling worthwhile. Throughout most of the range of  $F$  values, however,  $A_o$  is very close to 1. One then cannot look at the improvement  $I$  independently of the associated value of  $A_o$ , since a small  $I$  may lead to a sizable decrease in the total time the data base will be unavailable. For example, consider the situation when  $F = 20$  hours.  $I$  is only .015, but  $A_o$  is .9519, which means that  $A_2$  is .9662. Thus, the fraction of the time that the data base is unavailable decreases from 0.048 to 0.034. This translates into a nonnegligible decrease in downtime from 35 hrs./month to about 24 hrs./month.

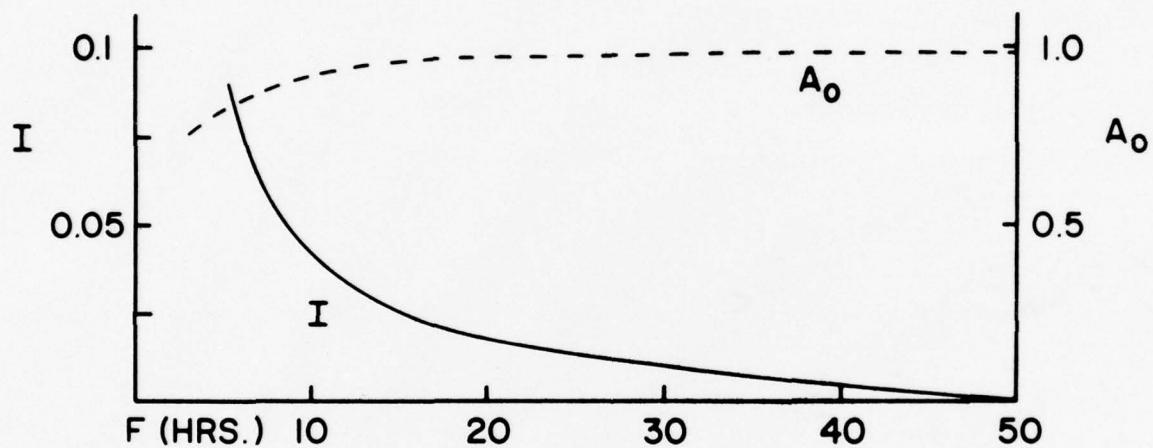


Figure 3  
 Single-site availability  $A_o$  and fractional improvement  $I$  through use of strategy 2.  
 Parameters are  $k = 0.01$ ,  $D = 0.01$  hr.,  $X = 1$  hr.,  
 $L = 0.5$  hr., and  $Y = F$ .

As a final comment on the remote journaling strategy described here, we note that availability may actually decrease as  $F$  increases. For example, suppose  $X = 2$ ,  $k = 0.25$ ,  $L = 0.5$  and  $D = 0$ . Then  $A_2 = .7692$  for  $F = 4$  and  $A_2 = .7647$  when  $F = 6$ . Differentiating  $A_2$  (for  $Y = F$ ) with respect to  $F$  shows that this decrease will occur whenever

$$k(k + 1)X > D + L.$$

Intuitively, this phenomenon occurs because for large  $k$  the effect of the lengthening audit trail to be processed outweighs that of the increasing reliability of the host computer.

Frequently updated remote journal. Clearly, there may be problems with the remote journaling strategy described in the last section because of the need to process an extremely long audit trail. Suppose, then, that we drop the assumption that  $Y = F$  and assume instead that the remote copy is periodically brought up to date. As an example, we might assume this updating to take place every two hours. Thus on the average the audit trail has been growing for 1 hour when the remote copy is activated. With all other parameters as specified for figure 3, but with  $Y = 1$ ,

$$I = .49/F.$$

This result is independent of  $k$  (because of the cancelling of the  $kX$  and  $kY$  terms), as long as  $k$  and  $F$  are such that the model is valid. The improvement is little different from what it was in the  $Y = F$  case. However, in appendix 2 we show that by decreasing  $Y$  we can considerably reduce the likelihood that the backup site fails before its copy is readied. Hence true availability will improve more than our simple formula indicates.

We do not include a graph of the result for  $Y = 1$ , since it would be almost identical to figure 3. To see why this should be so, consider more closely the formula for  $I$ .

$$I = \frac{A_2 - A_0}{A_0} = \frac{X + kX - D - L - kY}{F}$$

As long as  $k$  is small (or when  $X = Y$  as above) it is clear that

$$I \approx \frac{X - L}{F}.$$

Running spares. Here we assume that the backup copy is stored on disk for virtually instantaneous access and is kept almost up to date. Reasonable parameters for this case might be  $L = 0$ ,  $Y = .1$  hr., and (for comparison with the results above)  $X = 1$  hr.,  $k = .01$ . Then we have

$$I = \frac{0.999}{F}$$

We will not bother to graph this; this curve is again similar to that in figure 3, only now the values of  $I$  are approximately doubled. In this case, improvements of 5 to 10 percent are seen for  $F$  between 10 and 20 - certainly enough to make the strategy worthwhile. In fact, what happens in this case is that, under our assumptions, availabilities are brought up to very nearly unity. To see this, note that

$$A_2 = 1 - \frac{.01 + kY}{F + (1 + k)X} ,$$

and for our example  $kY = 0.001$ . Increasing  $k$  will cause somewhat smaller values of  $A_2$ , but  $A_2$  will be over 99 percent for a wide range of reasonable parameter values.

Effect of varying Y. We have looked at three separate cases which differ from one another in large part in the widely differing values for the parameter Y. To better understand the effect of this parameter, we select typical values of the other parameters (X = 1, L = 0.5, D = 0.01, F = 20) and consider  $A_2$  as a function of Y for several different values of k. When k = .01, we have

$$A_2 = 1 - \frac{0.51 + 0.01Y}{21.01}$$

The small coefficient of Y in this case makes the effect of Y minimal. As Y ranges between 0 and 20,  $A_2$  decreases linearly from 0.976 to 0.966. Now suppose that k is increased to 0.05. In this case as Y goes from 0 to 20,  $A_2$  decreases from 0.976 to 0.953. These are not very dramatic changes, although they will (as we noted above) be more impressive when translated into decreases in downtime. To a large extent, therefore, what makes the "running-spares" approach particularly worthwhile is not the small value of Y but the instantaneous access ( $L \approx 0$ ).

### Conclusions

We have presented here a model for data availability which, while superficial, does seem to reflect the realities of various strategies for backup. We have seen that remote journaling, in the sense of storing a copy in archival storage (e.g. tape) at a remote site, leads to availability improvement of at best 5 percent, which may be inadequate if single-copy availability is low. On the other hand, the running spares strategy, in which the remote copy is nearly up to date and almost immediately accessible, brings availability up to over 99 percent and appears to be worthwhile. It should be noted, however, that the running spares strategy is bound to be relatively expensive. Furthermore, before this strategy can be effectively used, many of the problems of multi-copy management must be solved. For example, updating must be synchronized in order to maintain consistency between the master and backup copies.

One final point should be made. In a sense, the gross availability of a data base is too vague a statistic. Suppose the availability is, say, 23/24. This might mean that approximately every 12 hours the data becomes unavailable for about a half hour. Or it might mean that once a month the data base disappears for more than a day. In a crisis, a half-hour delay in obtaining data might be tolerable but a one-day delay would not. The availability, then, must be looked at in conjunction with  $F$ , the mean time between failures.

References

Abramowitz, M. and Stegun, I.A. (editors)  
1964 Handbook of Mathematical Functions, National Bureau  
of Standards Applied Math. Series No. 55.

Barlow, R.E. and Proschan, F.  
1965 Mathematical Theory of Reliability, Wiley-Interscience.

Borgerson, B.R. and Freitas, R.F.  
1975 "A Reliability Model for Gracefully Degrading and  
Standby-Sparing Systems" IEEE Trans. Computers, C-24,  
pp. 517-525.

Chandy, K.M.; Browne, J.C.; Dissly, C.W.; and Uhrig, W.R.  
1975 "Analytic Models for Rollback and Recovery Strategies  
in Data Base Systems," IEEE Trans. Software Engineering,  
SE-1, pp. 100-110.

Chu, W.W.  
1973 "Optimal File Allocation in a Computer Network", in  
Computer-Communications Networks, N. Abramson and  
F. Kuo, eds., Prentice-Hall, pp. 82-94.

Frank, H.; Kahn, R.E.; and Kleinrock, L.  
1972 "Computer Communication Network Design - Experience with  
Theory and Practice," Proc. AFIPS National Computer Con-  
ference, AFIPS Press, Montvale, N.J., pp. 255-270.

Kleinrock, L.  
1975 Queueing Systems. Vol. 1: Theory, Wiley-Interscience.

Reynolds, C.H., and Van Kinsbergen, J.E.  
1975 "Tracking Reliability and Availability," Datamation,  
November 1975, pp. 106-116.

## Appendix 1

### Extension of Chu's Formula

In this appendix, we take up the question posed earlier as to data base availability when no time is lost in transferring usage (e.g., when downtime is scheduled at one site and transfer of usage to another site is prearranged). As we remarked earlier, a good way to study this problem would be through an extension of Chu's model [Chu, 1973].

Suppose  $n$  sites (all remote) have a copy of the data base, and that the availability of the  $i$ th site is  $a_i$ . (In general, this availability can be computed as the product of the availability of the  $i$ th host system times the availability of a communication link from the local site to site  $i$ .) Then the probability that site  $i$  is not available is  $(1 - a_i)$ , and the probability that none of the  $n$  sites is available is

$$U \equiv (1 - a_1)(1 - a_2)(1 - a_3) \dots (1 - a_n).$$

Hence the probability that at least one site is available is given by

$$A_s = 1 - U.$$

(Unlike Chu, we assume that we have no problem getting access to the network and so do not include Chu's factor for local host availability.)

To see how this gross availability is increased by the existence of multiple copies, consider the following examples.

1. Suppose all of the  $a_i$ 's are equal to 0.8. Then for  $n = 1$ ,  $A_s = 0.8$ ; but for  $n = 2$ ,  $A_s = 0.96$ ; and for  $n = 3$ ,  $A_s = 0.992$ .
2. Suppose that the data base is at site 1 where its availability is only 0.5. By placing a copy at a second site with availability 0.7, the overall availability becomes  $A_s = 0.85$ .

Finally, notice that if a copy of the data base is held locally, the formulas for  $A_s$  and  $U$  need not be changed at all. If we label the local site 1, then the value to be used for  $a_1$  is the availability of the data base through the local system. The fact that  $a_1$  does not involve network reliability as do the other  $a_i$ 's means that  $a_1$  may be slightly larger, but otherwise the formulation is unaffected.

## Appendix 2

### Stochastic Considerations

Basic assumptions. In the text of this paper we have been working exclusively with mean (or expected) values of parameters such as the time between site failures. As we indicated, however, site failure is a random process; not all questions can be answered by looking just at the mean time between failures. In particular, we have noted that our simplistic approach will predict too high an availability if, for example, hosts are likely to fail while the data base is being readied. In this appendix, then, we deal with some of these probabilistic questions in order to get a better understanding of the validity of the results we have computed.

First, recall that we are assuming the occurrence of failure to be a Poisson process. Essentially, this means that we assume that the probability that a failure occurs in any time interval from  $t_0$  to  $t_1$  is proportional to  $t_1 - t_0$ . Notice that if the constant of proportionality (which is just the failure rate) is  $1/F$ , then the mean time between failures is  $F$ , as we assumed earlier. The basic Poisson hypothesis also implies that the process is memoryless. That is, the probability of a failure in any time interval is independent of whether or when any failures occurred in the past.

One can then introduce a random variable  $Z$  giving the "time to failure" from an arbitrary starting point  $t = 0$ . The probability  $P\{Z \leq t\}$  that the machine fails before time  $t$  (i.e. in the time interval  $[0, t]$ ) can be shown to be

$$(A1) \quad P\{Z \leq t\} = 1 - \exp(-t/F).$$

Then the probability that the machine has not yet failed at time  $t$  is

$$(A2) \quad P\{Z > t\} = \exp(-t/F).$$

These simple formulas are adequate for computing most of the probabilities we are interested in.

Probability that backup fails before master is ready for use.

First, consider the following problem. What is the probability  $P_f$  that the backup may fail before the master site is again ready? We will first assume that  $X + kX$  is a constant. (The case where repair is also treated as a probabilistic process will be discussed below.)  $P_f$  is then calculated from equation (A1) with  $t = X + kX$ , the time required to repair and update the old master. We find that

$$(A3) \quad P_f = 1 - \exp(-(X + kX)/F).$$

For example, suppose that  $X = 2$  hours and  $k = 0.1$ . Then

$$P_f = 1 - \exp(-2.2/F).$$

Some values of this function are tabulated below. Values of  $F$  are given in hours. (Since only ratios are involved, the time units used do not matter as long as one is consistent.).

F	8	12	16	24	32	40	48
$P_f$	.24	.17	.13	.09	.07	.05	.04

The reader may well be dismayed that even for  $F$  (the mean time between failures) as large as 48 hours, there is still a 4% chance that the backup will fail before the old master is ready. This would leave a gap in availability which is not accounted for in our simplistic model. On the other hand, if the expected time to ready the master is considerably smaller - say  $X = 1$  hr.,  $k = 0.1$  - then  $P_f$  is only 0.09 for  $F = 12$ , 0.04 for  $F = 24$ , and 0.02 for  $F = 48$ . (Notice that

if  $(X + kX)/F$  is small,  $P_f$  is conveniently approximated by  $P_f \approx (X + kX)/F$ .)

If the  $P_f$  computed in any situation is large enough to seriously degrade availability, the solution is to provide a second backup, so that usage may be transferred to it if the first backup fails and the old master is not yet ready. This would, of course, not be worthwhile if the second backup requires so long to get ready that the old master will almost certainly be ready first.

We have investigated how these conclusions are affected by making the more realistic assumption that repair time is not a constant but also obeys some probability distribution. With this assumption, the probability  $P_f$  can be shown to be given by

$$(A4) \quad P_f = \int_{t=0}^{\infty} \exp [-(1 + k)t/F] W(t) dt,$$

where  $W(t)$  is the assumed probability density function for repair time.

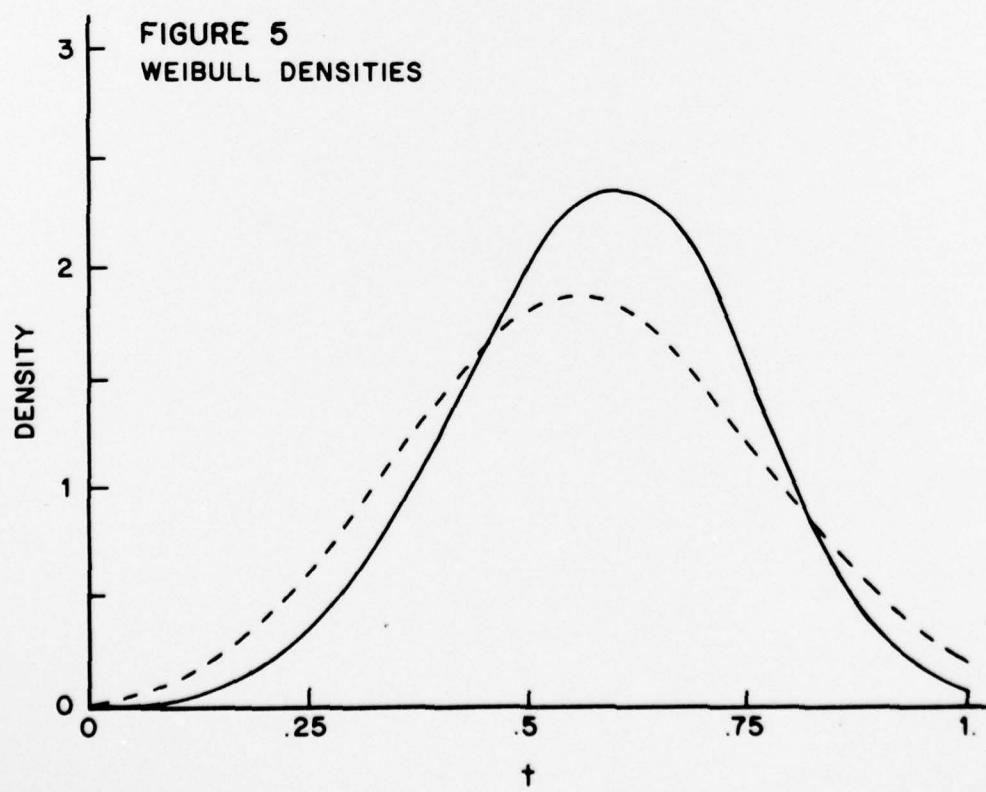
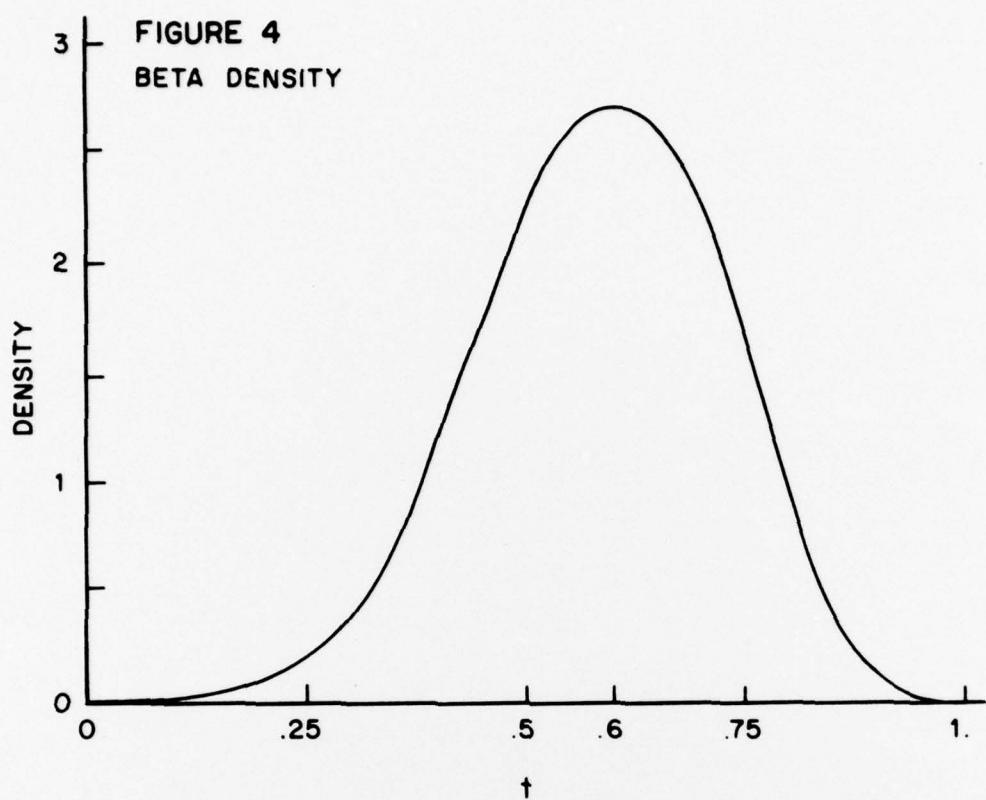
The only real difficulty in making this assumption is the apparent lack of the raw data which is needed before one can choose (and statistically validate) a  $W$ . From personal reports and from one study in the literature [Reynolds and Van Kinsbergen, 1975], we have put together the following general description of how repair time is distributed, at least for some systems.

1. The probability of repair within 15 minutes is essentially negligible.
2. The probability of repair within a half hour is something like 0.3 to 0.4.
3. In the vicinity of  $t = 0.5$  hr. the probability density curve rises sharply to its peak, so that the likelihood of repair within 45 minutes is between 0.8 and 0.9.

Notice that it should be relatively simple to obtain a good description of this sort for any particular system. All that is needed is a log of repair times.

There are two known probability distributions which have the right sort of shape to fit our general description of repair time. These are the Beta distribution [Abramowitz and Stegun, 1964; p. 930] and the Weibull distribution [Barlow and Proschan, 1965; ch. 2]. Both of these have two parameters -  $(\alpha, \beta)$  in standard notation - which can be used to adjust their precise shape. They differ in that the Beta distribution has  $W(t) = 0$  for  $t \leq 1$ , while for the Weibull distribution  $W(t)$  approaches zero exponentially as  $t \rightarrow \infty$ . Graphs of three such density functions which seem to describe repair time well are shown in figures 4 and 5. Figure 4 shows the density function for the Beta distribution with  $\alpha = 7$ ,  $\beta = 5$ . Figure 5 shows two Weibull distributions; the solid curve corresponds to  $(\alpha, \beta) = (6, 4)$  and the dashed curve to  $(\alpha, \beta) = (4, 3)$ . Note that the scale on the horizontal axis can be adjusted to fit longer (or shorter) expected repair times; i.e., "1" can be assumed an arbitrary time unit.

$P_f$  was computed from equation (A4) for 80 different combinations of distribution type (Beta or Weibull) and values of the parameters  $\alpha$ ,  $\beta$ ,  $F$ , and  $k$ . We observed that in no case did the value calculated differ by more than  $6 \times 10^{-4}$  from that calculated (using the appropriate mean value  $X$ ) from equation (A3). Since this discrepancy is of roughly the same magnitude as the truncation error in numerically computing the integral in (A4), we actually did not discover any difference between results computed from the two formulas. We conclude, therefore, that it is probably valid for all practical purposes to ignore the distribution of repair times and simply use the mean repair time  $X$  in equation (A3) to compute  $P_f$ .



Probability that backup site fails before its copy is ready

Next, what is the probability  $P_r$  that the backup site fails even before the backup copy can be gotten ready? This probability is given by

$$P_r = 1 - \exp(-(D + L + kY)/F).$$

(Again we assume that  $D + L + kY$  is a constant.) In our analysis of the remote journaling strategy, we assumed that  $Y = F$ , the mean time between failures. Let us also assume the nominal values  $L = 0.5$  hr.  $D = 0.01$  hr. and  $k = 0.1$ . With these values,

$$P_r = 1 - \exp(-(0.51 + 0.1F)/F).$$

Sample values are tabulated below.

F	12	16	24	36	48
P <sub>r</sub>	.13	.12	.11	.11	.10

Notice that as  $F$  becomes large,  $P_r$  approaches  $k$ . Unless  $k$  is very small,  $P_r$  is certainly not negligible. And the effect of a failure before the copy is readied could be serious. Again, the existence of a second backup would help, since it will seldom happen that both backup sites will fail before their copies can be readied.

In the discussion contained in the body of this report, we in fact concluded that for values of  $k$  as large as 0.1, a remote journaling strategy with no journaling taking place between failures is not worthwhile. We looked at the possibility that updating the remote journal more frequently might produce a more practical strategy. As the time,  $Y$ , that updates have been accumulating decreases,  $P_r$  will also decrease. Suppose that  $F = 24$  hrs.,  $L = 0.5$  hr.,  $D = 0.01$  hr. and  $k = 0.1$ . Then  $P_r$ , as a function of  $Y$ , behaves as follows:

Y	1	2	4	8	12	16	24
P <sub>r</sub>	.025	.029	.037	.053	.069	.084	.114

Thus when  $Y$  is small compared to  $F$ , the likelihood that the backup site fails before its copy can be readied is probably within acceptable limits.

Finally, consider the running spares strategy. In that case we assumed  $L = 0$  and  $Y = 0.1$ . If we again take  $k = 0.1$ , we find that

$$P_r = 1 - \exp(-0.02/F).$$

Here the probability that the backup site fails before the copy is ready is less than 0.2 percent as long as  $F$  is greater than ten hours. This small a figure will have a practically negligible effect on calculated availabilities.

### Appendix 3

#### Time to Process the Audit Trail

Elementary analysis. In this appendix we consider the question of how long it really takes to "catch up" when a backlog of updates has accumulated during the time a site is down. In the text, we have used Chandy's expression  $kY$  [Chandy et al., 1975], where  $Y$  is the length of time the updates have been accumulating, and  $k = u/b$ ,  $u$  being the rate of arrival of updates and  $b$  being the average rate at which updates are processed. (We assume that  $k < 1$ .) However, it is clear that during the time interval  $kY$  more updates are accumulating, and it takes an additional time  $k^2Y$  to process these. Continuing to add on these correction terms, we generate the infinite series

$$(k + k^2 + k^3 + \dots)Y$$

as a better formula for the catch-up time  $T_c$ . Computing the sum, we get

$$T_c \approx kY/(1 - k).$$

This is a slightly larger number than the first estimate  $kY$ , but the difference is a small percentage for the expected range of  $k$  values. (Chandy et al. state that "values of  $k$  of order 1/10 or less are to be expected.")

Analysis using queueing theory. Even this analysis, however, appears to be too simplistic. The arrival and processing of updates should really be modeled by a single-server queueing system. If the arrivals form a Poisson process (arrival rate  $u$ ) and processing time is exponentially distributed (with mean processing rate  $b$ ), the queueing system is one which can be analyzed. In queueing theory, the quantities of interest are  $P_n(t)$ , the probability that there are  $n$  items (updates,

in our case) in the queue at time  $t$ . After the system has been running for a while, the probabilities  $P_n(t)$  will approach equilibrium (or steady state) values  $p_n$ . Of course, equilibrium is never actually reached if the initial distribution is not the equilibrium one. But it does make sense to describe the catch-up time  $T_c$  as the time to reach approximate equilibrium.

Fortunately, both the time-dependent and the equilibrium distributions that we need are available in the literature [Kleinrock, 1975], so that we have at hand the information we need to investigate the approach to equilibrium.

The equilibrium distribution is given by

$$p_n = (1 - k)k^n.$$

The time-dependent distribution we are interested in corresponds to having some number  $i$  of updates in the queue at time  $t = 0$ . That is, we have the initial conditions

$$\begin{aligned} p_n(0) &= 1 & \text{for } n = i; \\ p_n(0) &= 0 & \text{for } n \neq i. \end{aligned}$$

Kleinrock [1975; p. 77] gives the solution of this problem as

$$\begin{aligned} p_n(t) &= \exp(-(u + b)t) [k^{(n-i)/2} I_{n-i}(at) \\ &\quad + k^{(n-i-1)/2} I_{n+i+1}(at) \\ &\quad + (1 - k)k^n \sum_{j=n+i+2}^{\infty} k^{-j/2} I_j(at)], \end{aligned}$$

where  $a = 2uk^{1/2}$ , and  $I_j$  (standard notation) is the modified Bessel function of the first kind of order  $j$ .

In order to study the approach to equilibrium, the following formulas for Bessel functions are needed:

$$1) \text{ As } z \rightarrow \infty, I_j(z) \sim \{e^z/(2\pi z)^{1/2}\} \{1 - \frac{4j^2 - 1}{8z} + \dots\}$$

$$2) \sum_{j=-\infty}^{\infty} k^{j/2} I_j(z) = \exp[(z/2)(k^{1/2} + k^{-1/2})].$$

$$3) I_j(z) = I_{-j}(z).$$

$$4) e^z = I_0(z) + 2I_1(z) + 2I_2(z) + \dots$$

[Abramowitz and Stegun, 1964; p. 374 ff.]

Using 2) and 3), we find that

$$\sum_{j=-\infty}^{\infty} k^{-j/2} I_j(at) = \exp(t(u + b)).$$

Noting that the infinite summation in the expression for  $P_n(t)$  contains only a portion of these terms, we use 4) to show that the summation over the negative powers is negligible, and use 1) to estimate the finite number of missing positive powers. That is, for large  $t$  we make the approximation

$$\sum_{j=n+i+2}^{\infty} k^{-j/2} I_j(at) \approx \exp(t(u + b)) - \frac{\exp(at)}{(2\pi at)^{1/2}} \sum_{j=0}^{n+i+1} k^{-j/2} \left\{ 1 - \frac{4j^2 - 1}{8at} \right\}$$

(The  $j^2$  term in the asymptotic expression is needed since unfortunately the constant terms cancel in the ultimate expression for  $P_0(t) - p_0$ .) Furthermore, we notice that, when we substitute the asymptotic formulas for  $I_j(at)$  into  $P_n(t)$ , we obtain products of exponentials

$$\exp(-(u + b)t) \exp(at)$$

which simplify to

$$\exp(-(u^{1/2} - b^{1/2})^2 t).$$

After considerable algebraic manipulation, we obtain

$$P_0(t) - p_0 \sim \frac{\exp(-(u^{1/2} - b^{1/2})^2 t) ik^{-i/2}}{2\sqrt{\pi} \sqrt{ub} t^{3/2} (1 - k^{-1/2})}.$$

Recall that we have said that the catch-up time  $T_c$  should be the time at which the  $P_n(t)$ , and in particular  $P_0(t)$ , are "close" to their equilibrium values. This would mean that the right side of the above expression should be small. However, notice that the exponential factor decreases rapidly with  $t$ , while the factor  $ik^{-1/2}$  is very large if the queue is long at time  $t = 0$ . To a good approximation, then, we can assume that equilibrium is reached when these factors approximately cancel; that is, when

$$\exp(+ (u^{1/2} - b^{1/2})^2 t) \approx ik^{-1/2}.$$

Taking logarithms, we obtain the following formula for  $T_c$ :

$$T_c = \frac{-i\ln k + 2\ln i}{2(u^{1/2} - b^{1/2})^2}$$

If the term  $2\ln i$  is neglected, this expression simplifies in an interesting way. Suppose updates have been accumulating for a time period of length  $Y$ . Then  $i = uY$ , and

$$\begin{aligned} T_c &= \frac{-uY\ln k}{2b(k^{1/2} - 1)^2} \\ &= kY\{(-\ln k)/2(1 - k^{1/2})^2\}. \end{aligned}$$

The quantity in brackets has the curious property of lying very close to 2.5 for  $k$  between 0.04 and 0.15. (For  $k = 0.01$  it is still 2.84; although it grows rapidly thereafter as  $k$  decreases.) Notice also that adding on the term  $2\ln i$  will serve to increase the effective  $T_c$ . On the other hand, taking account of the terms in the denominator of the approximate expression for  $P_0(t) - p_0$  (in particular the factor  $t^{3/2}$ ) will serve to decrease it. It seems not unreasonable, then, to claim as we did early in the paper, that  $T_c \approx 2kY$ .

It should be emphasized that the queueing theory analysis above depends strongly upon the assumption that updates are arriving randomly - i.e., that the arrivals form a Poisson process. If arrivals are instead bunched up during certain time periods, results may be quite different. For example, if a number of updates  $uY$  have accumulated and must be processed during a time when no new updates are arriving, then clearly  $uY/b = kY$  will be the correct expression for  $T_c$ . On the other hand, if the backlog of updates must be processed during a time period when a particularly large number of new updates are being entered, then  $T_c$  will be greater than the queueing analysis has indicated.

## Appendix 4

### Sensitivity to Parameter Values

In any model, it is useful to determine how sensitive the output values are to changes in the inputs. Obviously, the inputs are only known approximately or are statistical averages. If the output changes drastically for a small change in an input value, the model is rather useless for predictive or decision purposes. Chandy et al. [1975] use the elasticity  $E(f,y)$ , essentially the "percentage change in  $f$  caused by a percentage change in  $y$ ", to investigate the sensitivity of a function  $f$  with respect to a parameter  $y$ . Formally,  $E$  is defined by

$$E(f,y) = \frac{\partial f}{\partial y} \cdot \frac{y}{f}.$$

We have investigated the elasticity of  $U = 1 - A_2$  with respect to all of the input variables. (Working with  $U$  instead of  $A_2$  simplifies the algebra without changing the conclusion.) We find that for all parameters

$$\left| \frac{\partial U}{\partial y} \cdot \frac{y}{U} \right| < 1.$$

For example, taking  $y = k$ ,

$$\frac{\partial U}{\partial k} = \frac{FY + XY - DX - LX}{(F + X + kX)^2}, \text{ and}$$

$$\begin{aligned} \left| \frac{\partial U}{\partial k} \cdot \frac{k}{U} \right| &= \left| \frac{k(FY + XY - DX - LX)}{(F + X + kX)(D + L + kY)} \right| \\ &= \left| \frac{kFY + kXY - \dots}{kFY + kXY + \dots} \right| < 1. \end{aligned}$$

And for  $y = Y$ ,

$$\left| \frac{\partial U}{\partial Y} \cdot \frac{Y}{U} \right| = \frac{kY}{F + X + kX} \cdot \frac{F + X + kX}{D + L + kY} = \frac{kY}{D + L + kY} < 1.$$

Similar computations show that the elasticities of  $U$  with respect to  $D$ ,  $L$ ,  $X$ , and  $F$  are all less than one. Elasticities of  $U$  are connected to those of  $A_2$  through

$$\left| \frac{\partial A_2 \cdot y}{\partial y} \frac{y}{A_2} \right| = \left| \frac{\partial U}{\partial y} \frac{y}{A_2} \right| < \left| \frac{\partial U}{\partial y} \frac{y}{U} \right|,$$

as long as  $A_2 > U$ . We may conclude therefore that our model is stable, being relatively insensitive to small changes in parameter values.

## UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ⑧ CAC Document Number 181 CCTC-WAD Document Number 6501	2. GOVT ACCESSION NO. ⑨ 19	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Research in Network Data Management and Resource Sharing. The Effect of Backup Strategy on Data Base Availability.		5. TYPE OF REPORT & PERIOD COVERED Research Report, ⑨
6. AUTHOR(s) Geneva G. Belford, Paul M. Schwartz, and Suzanne Sluizer		7. PERFORMING ORGANIZATION NAME AND ADDRESS Center for Advanced Computation University of Illinois at Urbana-Champaign Urbana, Illinois 61801
8. PERFORMING ORGANIZATION NAME AND ADDRESS Command and Control Technical Center WWMCCS ADP Directorate 11440 Isaac Newton Sq., NO., Reston, Va. 22090		9. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ⑩ 15 DCA100-75-C-0021
10. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) ⑪ UIUC-CAC-DN-76-181, CAC-181		11. REPORT DATE ⑪ 12 Feb 1976
		12. NUMBER OF PAGES ⑫ 46p.
		13. SECURITY CLASS. (of this report) UNCLASSIFIED
		14. DECLASSIFICATION/DOWNGRADING SCHEDULE
15. DISTRIBUTION STATEMENT (of this Report) Copies may be obtained from the National Technical Information Service Springfield, Virginia 22151		
16. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) No restriction on distribution		
17. SUPPLEMENTARY NOTES None		
18. KEY WORDS (Continue on reverse side if necessary and identify by block number) information system modeling data base availability data base backup distributed data management		
19. ABSTRACT (Continue on reverse side if necessary and identify by block number) Formulas are developed for the study of improvements in data base availability due to the existence of a backup copy located at an alternate site in a network. Several backup strategies are compared.		

407227

<b>BIBLIOGRAPHIC DATA SHEET</b>		1. Report No. UIUC-CAC-DN-76-181	2.	3. Recipient's Accession No.
4. Title and Subtitle  Research in Network Data Management and Resource Sharing - The Effect of Backup Strategy on Data Base Availability		5. Report Date February 1, 1976		
6.				
7. Author(s)  Geneva G. Belford, Paul M. Schwartz, and Suzanne Sluizer		8. Performing Organization Rept. No. CAC #181		
9. Performing Organization Name and Address  Center for Advanced Computation University of Illinois at Urbana-Champaign Urbana, Illinois 61801		10. Project/Task/Work Unit No.  DCA 100-75-C-0021		
12. Sponsoring Organization Name and Address  Command and Control Technical Center WWMCCS ADP Directorate 11440 Isaac Newton Square, North Reston, Virginia 22090		13. Type of Report & Period Covered Research		
14.				
15. Supplementary Notes				
16. Abstracts  Formulas are developed for the study of improvements in data base availability due to the existence of a backup copy located at an alternate site in a network. Several backup strategies are compared.				
17. Key Words and Document Analysis. 17a. Descriptors  information system modeling data base availability data base backup distributed data management				
17b. Identifiers /Open-Ended Terms				
17c. COSATI Field/Group				
18. Availability Statement No restriction on distribution Available from the National Technical Information Service, Springfield, VA 22151		19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 44	
		20. Security Class (This Page) UNCLASSIFIED	22. Price	